

Introduction to TCP/IP & Embedded Internet (1)

learn about remote control via the Internet

By P. Stuhlmüller

What has been for some time possible with a 'big' system can now also be realised with a small system: microcontrollers have become Internet-capable, since they have learned how to 'speak' TCP/IP. In this article, the author first describes a few important fundamentals of Internet communications that are relevant to our target application.

Embedded Internet

This is the first of a three-part series. The first article, in this issue, discusses the fundamentals of Internet communications, the second article (which will appear in next month's issue) deals with programming a TCP/IP stack, and the series will culminate in the Summer Circuits issue (July/August) with the first 'Elektor Electronics Internet Project'. The subject of this project is a universal mini-webserver (Web-I/O) with the following noteworthy features:

- 80186 processor core
- integrated Ethernet interface
- 2 serial Ports
- I²C interface
- 16 digital inputs
- 16 digital outputs
- liquid crystal display
- expansion connector for custom hardware development
- DOS-like multitasking operating system
- TCP/IP stack with PPP, HTTP, FTP, SMTP and Telnet
- many free sample programs available on the Internet
- easily programmed using various DOS compilers

Don't miss this special project!

For years, the control and monitoring of complex systems and processes from a central control station has been a standard and proven practice. In the present era of high-speed global networking, the increasingly

smaller role played by spatial separation can be regarded as fully logical. Nowadays, all-encompassing data networks at local and global levels are part of the infrastructure of

every modern industrial society. For a long time already, data lines have been used to link travel agencies to airlines, ticket agencies to promoters and information systems to their operators, although this is not always externally evident. Communications satellites and terrestrial transmitters are also controlled remotely, of course, and many industrial and research facilities would not be conceivable without remote control and remote maintenance, due to the dangers that their emissions pose to humans. The list of systems that are dependent on long-distance data exchange could be extended almost at will. The methods and processes of data communications are generally adapted to their intended uses, and transmission security is often an important consideration in the implementation of the transmission paths.

The idea that the worldwide data and information transmission medium called the Internet is also suitable for

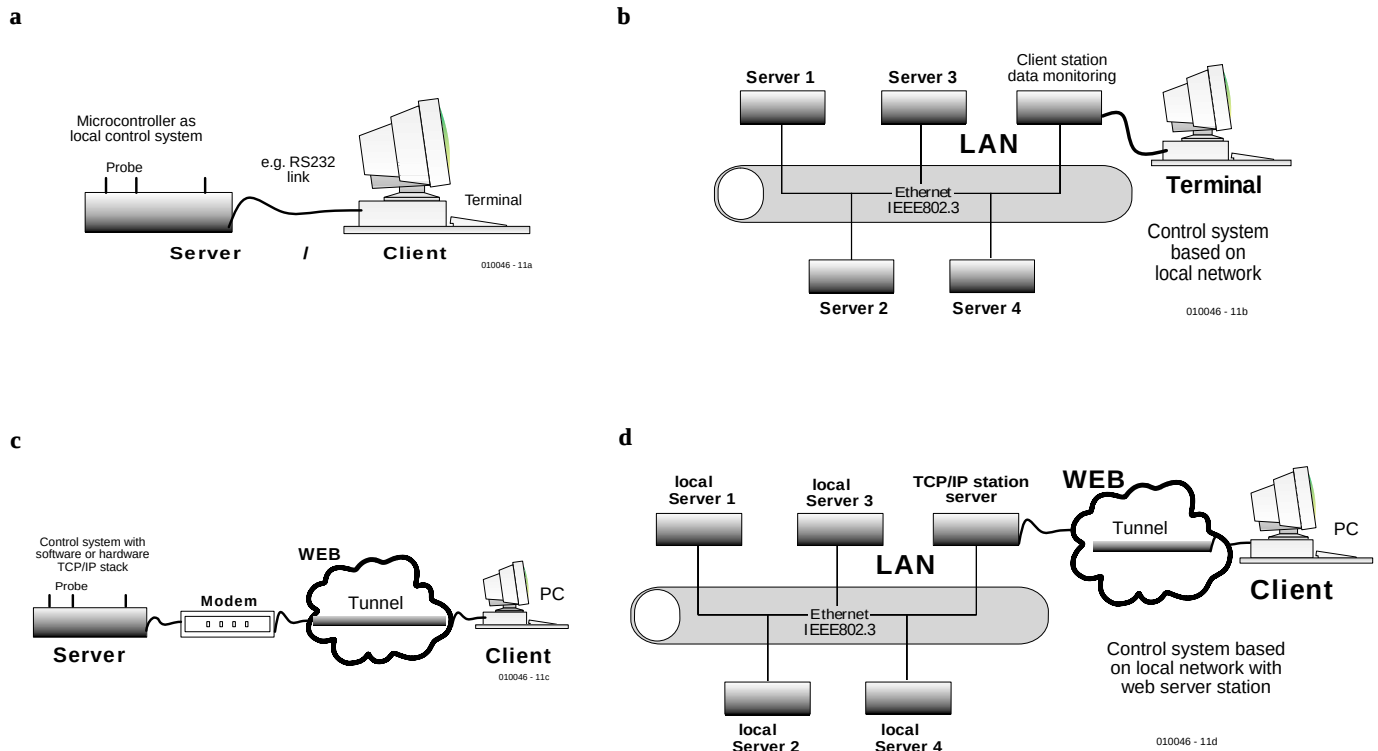


Figure 1. Client-server system as an isolated system (a), as part of a LAN (b), as a Web-capable individual system (c) and as part of a Web-capable LAN (d).

many remote control and monitoring tasks is certainly not new; in fact, such applications were envisaged at the very beginning. The systems that are supposed to communicate with each other via the Internet only have to speak the language of the Internet: they must master a 'stack' of protocols that is collectively referred to as 'TCP/IP'. In the rest of this article, we take a closer look at the individual aspects of TCP/IP that are important in our context.

Client-server model

The meaning of the term 'client-server model', which originates from network technology, can be quickly explained using a practical example. Let's assume that we have a measurement system that must be operated from a more or less remote location and that the measured values (data) must be evaluated at this remote location. Here the measurement system is the server (data supplier) and the control and evaluation facility is the client (data consumer). Four different types of client-server configuration are sketched in **Fig-**

ure 1. The simplest type, as shown in Figure 1a, actually needs no further explanation. In Figure 1b, several servers are networked together by means of a local area network (LAN), and they are collectively linked to the control and evaluation centre (with the client). Figures 1c and 1d correspond to the configurations shown in Figures 1a and 1b, with the decisive difference that here the transmission paths run through the Internet in the form of 'tunnels'.

A measurement data system that communicates via the Internet, and in which Internet capability is already present in the system itself (in other words, this capability is 'embedded' in the system), has the following advantages:

- It is possible to access the system via the Web from every corner of the world (and even from outer space).
- Without regard to place and time, the system can be monitored and controlled by specialists who only have to concentrate on these tasks.
- Security mechanisms, such as data encryption and passwords, pre-

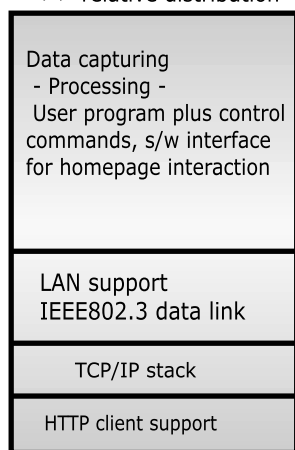
vent unauthorized access and improper evaluation of the data.

- At the operator interface level, the system behaves exactly as though it were installed locally.
- Naturally, there are also a few disadvantages:
- The cost and complexity of the software for the system are significantly higher than for a locally installed system.
- The implementation of Internet capability is generally only possible with high-performance hardware, which significantly increases costs.
- Data transport via the Internet results in operating costs (provider charges) that would not accrue with a local installation.

Paths through the Internet

In order to be able to transport measurement data or any other desired data via the Internet, the data must be packaged into suitable 'shells'. This is necessary if only because the transport paths in the Internet are utilised by many servers and clients *in common*, for predominantly economic reasons. The data thus do not travel as a single bundle; instead, they travel in small individual packets from the server to the client via a relatively large or small number of intermediate stations. The

OS structure for web server
* >> relative distribution



010046 - 12

Figure 2. Software modules of a mini-webserver.

requirements document that defines how the utility data are to be split into partial shipments and packaged is called TCP/IP, or in full, 'Transport Control Protocol / Internet Protocol'.

Figure 2 shows schematically how the individual tasks are distributed in a client-server data acquisition system that communicates via the Internet. Aside from the hardware and software for acquiring and preparing the measurement data, it is necessary to have a LAN support module, which primarily coordinates the local networking of the measurement equipment, a TCP/IP support module

that enables data transport via the Internet, and an HTTP support module that looks after additional necessary or desirable functions. Since the measurement data are divided into packets that travel along the data highway along with many other unrelated packets. Each packet must be identified by an unambiguous and unique recipient address. For both the client and the server, the handling and processing of addresses is a key task that must be mastered by every system that communicates via the Internet.

Internet addresses

Every Internet address consists of four blocks of digits in which the numbers from 0 to 255 may occur, which corresponds to a word width of 32 bits. The address is composed of two parts: the network address, which is the same for all stations connected to a single network, and the individual host address. Internet addresses are divided into five classes, labelled A through E, which have different schemes for splitting the available 32 bits into the network address and the host address. **Table 1** summarises these classes.

For local networks that are constantly connected to the Internet, the address must be registered with InterNIC (the Internet Network Information Center). Other types of Inter-

net users, who only intermittently communicate via the Internet (using a modem connection, for example) can be assigned an address dynamically, which means for the duration of the connection.

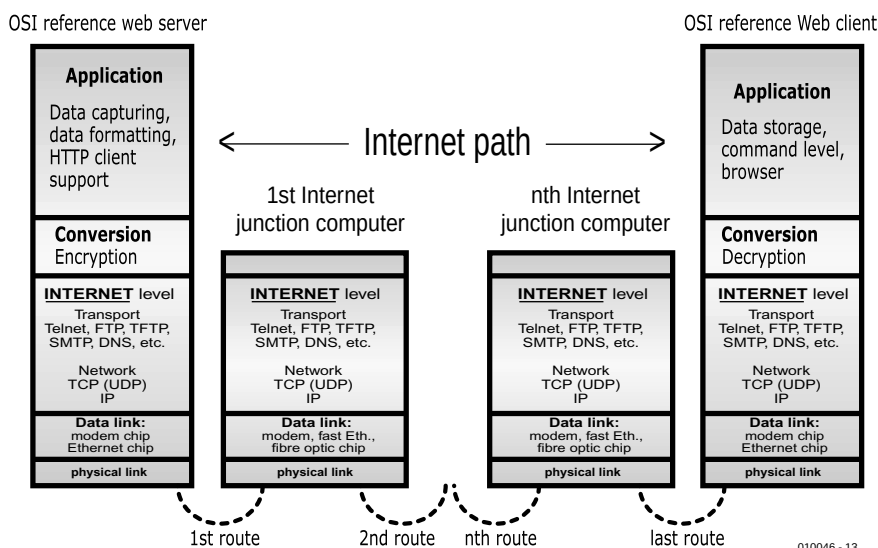
TCP/IP service levels

The design of the software structure of a client-server Web model boils down to a division into a number of service levels. The Reference Model, which has been given the name 'Open System Interconnection' (OSI), is a representation of the software modules that are mandatory for a client-server Web model. In the case of a mini-webserver, the Reference Model reflects all of the activities that the microcontroller must carry out in order to feed the utility data into the Internet via a modem or other data link in the proper form for the Internet. This structure is illustrated in detail in **Figure 3**. The TCP/IP module has the most important role for data transport via the Internet. It adds the recipient address to the data packets that are to be sent, and in addition it adds 26 values that have a prescribed sequence and represent various delivery features. The structure shown in Figure 3 is vertically oriented, so it runs from the top to the bottom.

Another important aspect of the client-server Web model is the communications management function, which for example governs whether the client or the server initiates communications. Normally, the role of the 'caller' falls to the client, for specific reasons (e.g. protection against manipulation), but the role assignment can also be reversed. Of course, this must first be implemented in the software.

Into and through the Net

The computers that are involved in the Internet, and their operating systems, belong to a wide variety of different worlds: in addition to DOS and Windows computers, Unix, SunOS and AS400 systems, to name only a few, there are also large computers, some of which are familiar only to specialists. Independent of the type and size of the system, however,



010046 - 13

Figure 3. OSI Reference Model for Web-based control systems.

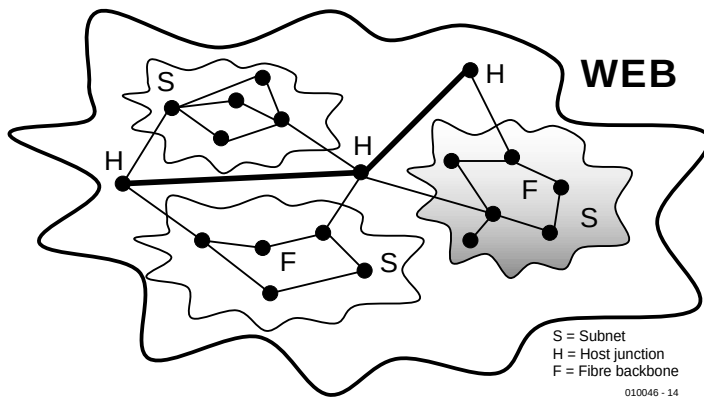


Figure 4. Schematic representation of data transfers in the Internet.

Table 1. Division of the Internet addresses in groups A through F

Class	Network	Hosts	Address range
A	1 – 126	16 777 214	0.0.0.1 to 127.255.255.255
B	128 – 191	65 534	128.0.0.0 to 191.255.255.255
C	192 – 223	254	192.0.0.0 to 223.255.255.255
D	224 – 239		Reserved for multicast addresses.
E	240 – 254		Reserved; presently unused.

many procedures are the same for all of them. A microcontroller-based mini-webserver with its own operating system connects to the Internet in exactly the same manner as a mainframe machine, to the extent

that the *input conditions* are the same. Both systems must go through the same log-on procedures if (for example) they are linked to the external data world via the So bus of an ISDN connection.

The access procedure via the ISDN So bus is standardised as follows:

- A connection to the node computer of the provider is set up using the Telnet network protocol. This connection initially has the nature of a terminal connection.
- The data packet is sent to the node computer using the platform-independent FTP protocol (File Transfer Protocol). This link is a point-to-point connection.
- From this point onwards, TCP/IP (or more precisely, IP) is involved in the data communications. The node computer of the provider evaluates the file header and specifies the path to the next node computer. This routing function is looked after by a software module that works with routing tables. The routing tables can be generated manually (static routing), but they can also be continually updated to match actual circumstances (dynamic routing). Besides this, it is possible to have the routing be based exclusively on hardware. The 'IP tunnelling' method is used for transporting data packets from one node computer to the next one.
- At latest after the packet has been handed over to the first node computer, a confirmation is sent back to the sending computer. The processes in the node computer are rather complex, so here we will describe only the most important feature: for every possible path to the recipient, a maximum data packet length is determined, and this is a component of the input conditions for the link in question within the network. This is because each link has a limitation on the maximum length of data packets that can be transmitted, which means that if the data packet is too long, it must be split into smaller pieces. Since the transmission speeds of the various paths through the network that are employed can vary considerably, there is no guarantee that the partial packets will arrive in the correct order at the recipient. It is the task of the recipient software to reconstruct the original file from the partial packets.

For clarification, a section of the Internet is sketched out in **Figure 4**.

TCP/IP header

From Tables 2a and 2b, we can see which parameters are contained in the TCP and IP parts of the file header. All of the header words have a width of 32 bits, and they are arranged in the file header so that they can be evaluated as quickly as possible. The programmer of a client-server Web system must without exception use the values in the file header that are applicable to the system that he or she has developed! If the client-server

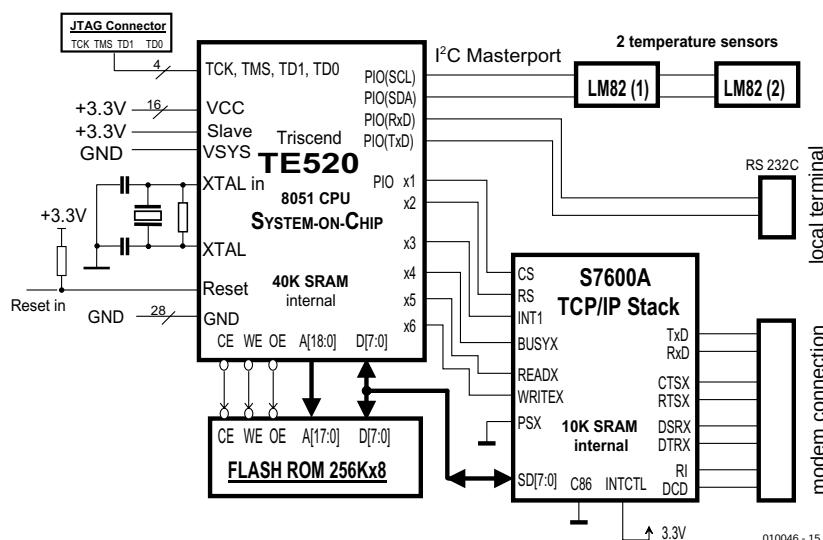


Figure 5. Basic circuit diagram of an isolated mini-webserver using the Seiko S7600A as the TCP/IP hardware stack.

Table 2a. TCP entries

Field Designation	Length (bits)	Meaning
Source Port	16	Sender address of the process or data service of a higher-level protocol. Together with the IP address, pointer to a procedure socket. Data packet at the Telnet port of the recipient. The Telnet port communication sender/recipient is NOT a component of the Sender IP address.
Destination Port	16	Socket process or data service of the recipient computer. The sockets on the sender and recipient sides remain active as long as the connection exists.
Sequence Number	32	Data flow; sequence number of the first byte in the data packet.
Acknowledge Number	32	Data flow; confirmation of the receipt of all sent bytes.
Data-Offset	5 – 15 usually 5	Number of 32-bit words in the TCP from this word onward, including the data set (always a multiple of 32 bits). Reserved 6 For future applications.
Condition-Control-Flags	6	URG: urgent pointer; ACK: the Acknowledge number means important/unimportant (1/0); PSH: handover to a higher-level protocol (yes/no); RST: terminate existing link (yes/no); SYN: to the recipient: set up connection (yes/no); FIN: sender indicates the end of the connection.
Windows-Limit/Size	16	Flow control between the sender and the recipient of the data block in the original datagram.
Checksum	16	Checksum of the 96-bit pseudo-header (32 + 32 + 8 + 8 + 16 bits) IPSA + IPDA + empty protoident + TCP segment.
Urgent-Pointer *	16	Positive offset of the sequence number.
Service-Options **	Var.	variable Type, length and options data in the datagram.
Padding	8	Fill data to ensure that the header has a 32-bit format.
Data Set	32	

* Indicates that the TCP data are highly urgent. The pointer always points to the end of prior data. If the value of the pointer is added to the sequence number, the result is the sequence number of the last urgent TCP data set. Data of this sort are interrupts of breaks that are transmitted by an opposite party.

** Field length; depends on the type, length and special data. Every TCP implementation must support all defined TCP options.

Table 2b. IP entries

Field Designation	Length (bits)	Meaning
Version	4	Version 4 IP header version, currently 4.
IHL	4	Internet Header Length.
Service Type	8	IP datagram services: delay, throughput, reliability.
Total Length	16	Datagram length, IP header + data block.
Identification	16	Identifier; assignment of the fragment in the datagram.
Flags	3	DT: fragment, MF: last fragment / no fragment. Fragment Offset 13 Position of the data fragment relative to the start of the data block in the original datagram.
Time to Live	16	Remaining lifetime of the datagram.
Protocol *	16	ICMP, IGMP, TCP, EGP, UDP, XNS-IDP, ISO-IP, OSPFIGP
IP Header Checksum	16	Checksum for the IP Header (ones-complement).
IP Source Address	32	Source address of the server (sender).
IP Destination Address	32	Target address of the client (recipient).
Options **	7 min.	Classes: Control and Debugging, IT Timestamp.
Padding	8 - 24	Fill data to ensure that the header has a 32-bit format.

* ISO IP (ISO Internet Protocol), UDP (User Datagram Protocol).

** Copy Flag in the fragmentation process.

Web system is a component of a local area network (LAN), then normally only the station that sets up the link to the Internet has to run TCP/IP. If all stations were to have direct access to the Internet, the software overhead would be quite significant.

TCC/IP in a chip

Everything that is necessary for a typical mini-webserver has already been illustrated in Figure 2. For an isolated mini-webserver, the necessary functions could basically be implemented using a suitable microprocessor and the associated software. An interesting alternative to this classical solution is provided by Seiko Instruments in the form of an IC that renders programming a TCP/IP stack redundant. Inside the S7600A chip, the TCP/IP stack is

already 'cast' in hardware in a ready-to-use form. Here Seiko Instruments has implemented the current version (Version 4) of the TCP/IP protocol suite in an IC with UDP (User Datagram Protocol) and PPP (Point-to-Point Protocol). One limitation that should be mentioned is that UDP is somewhat simpler than TCP and not quite as reliable, although the working speed of UDP is higher.

The S7600A incorporates an integrated serial interface (UART), which can be connected to a modem via an RS232 line driver. The 8-bit parallel port of the S7600A is compatible with standard microcontrollers, and if necessary it can easily be adapted. Figure 5 shows the basic diagram of a mini-webserver using the S7600A. The TE520 is a configurable microcontroller with an 8051 core, 40 kB of SRAM and 2048 freely programmable logic cells. Two LM85 tempera-

ture sensors are shown connected to the I²C bus by way of example. What they measure can be queried from continents away via the Internet!

Another solution is based on the well known Zilog Z80 microprocessor, which nowadays can work at clock frequencies up to 80 MHz, can address up to 16 MB of peripheral and memory space and also offers on-chip DSP functions. With this device, the associated 'Embedded Web Server Software Suite' and a Z02915 single-chip modem, we have a mini-webserver with astonishing capabilities.

Among the other current hardware solutions, the Intel SA1110 should be mentioned. With a rich assortment of on-chip peripherals, this IC together with a V.90 modem chipset is a powerful alternative solution.

In the next instalment of this series the focus is on programming a typical mini-webserver, while in the third instalment the mini-webserver will be presented as an *Elektor Electronics* construction project.

(010046-1)